Docket No. 247558US2RD

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Kensaku YAMAGUCHI, et al. GAU:

SERIAL NO: New Application EXAMINER:

FILED: Herewith

FOR: DATA ACCESS CONTROL METHOD FOR TAMPER RESISTANT MICROPROCESSOR USING CACHE MEMORY

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

☐ Full benefit of the filing date of U.S. Application Serial Number , filed , is claimed pursuant to the provisions of **35 U.S.C. §120**.

☐ Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of **35 U.S.C. §119(e):** **Application No.** **Date Filed**

■ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of **35 U.S.C. §119**, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

COUNTRY	APPLICATION NUMBER	MONTH/DAY/YEAR
Japan	2003-012558	January 21, 2003

Certified copies of the corresponding Convention Application(s)

■ are submitted herewith

☐ will be submitted prior to payment of the Final Fee

☐ were filed in prior application Serial No. filed

☐ were submitted to the International Bureau in PCT Application Number
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.

☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and

☐ (B) Application Serial No.(s)

- ☐ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Marvin J. Spivak
Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124

Customer Number
22850
Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 05/03)

T646

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　１月２１日

出願番号
Application Number: 特願２００３－０１２５５８

[ST. 10/C]: [ＪＰ２００３－０１２５５８]

出願人
Applicant(s): 株式会社東芝

２００３年　７月１８日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号　出証特２００３－３０５７４７７

【書類名】　特許願
【整理番号】　13B029038
【提出日】　平成15年 1月21日
【あて先】　特許庁長官殿
【国際特許分類】　G06F　5/00
【発明の名称】　耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法
【請求項の数】　14
【発明者】
【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
【氏名】　山口　健作
【発明者】
【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
【氏名】　橋本　幹生
【特許出願人】
【識別番号】　000003078
【氏名又は名称】　株式会社　東芝
【代理人】
【識別番号】　100083806
【弁理士】
【氏名又は名称】　三好　秀和
【電話番号】　03-3504-3075
【選任した代理人】
【識別番号】　100068342
【弁理士】
【氏名又は名称】　三好　保男

【選任した代理人】

【識別番号】　100100712

【弁理士】

【氏名又は名称】　岩▲崎▼　幸邦

【選任した代理人】

【識別番号】　100100929

【弁理士】

【氏名又は名称】　川又　澄雄

【選任した代理人】

【識別番号】　100108707

【弁理士】

【氏名又は名称】　中村　友之

【選任した代理人】

【識別番号】　100095500

【弁理士】

【氏名又は名称】　伊藤　正和

【選任した代理人】

【識別番号】　100101247

【弁理士】

【氏名又は名称】　高橋　俊一

【選任した代理人】

【識別番号】　100098327

【弁理士】

【氏名又は名称】　高松　俊雄

【手数料の表示】

【予納台帳番号】　001982

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法

【特許請求の範囲】

【請求項１】　暗号化されたプログラムの実行コードの読み出し要求を処理するキャッシュメモリ制御手段と、

前記キャッシュメモリ制御手段からの復号化要求に基づき、前記実行コードを記憶装置より読み出し、所定の暗号鍵によって復号化する復号化手段と、

前記復号化手段にて復号化した前記実行コードを記憶するキャッシュメモリとを具備し、

前記キャッシュメモリ制御手段は、

前記実行コードの読み出し要求を処理する際に、前記キャッシュメモリ内に前記実行コードが存在し、かつ該実行コードが前記所定の暗号鍵と同一の暗号鍵によって復号化された実行コードであった場合には、前記復号化手段へ前記復号化要求を出す代わりに、前記キャッシュメモリ内の実行コードを使用するよう制御することを特徴とする耐タンパマイクロプロセッサ。

【請求項２】　前記実行コードは、復号化後に前記キャッシュメモリ内のキャッシュライン毎に記憶され、前記キャッシュラインは前記実行コードの復号化に用いた暗号鍵を格納するための秘密保護属性保持部を有することを特徴とする請求項１に記載の耐タンパマイクロプロセッサ。

【請求項３】　前記暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵が更新されて記憶される鍵値レジスタを備え、

前記キャッシュメモリ制御部は、前記キャッシュメモリ内の実行コードが前記所定の暗号鍵と同一の暗号鍵によって復号化された実行コードである否かを判断する際、前記鍵値レジスタと、実行しようとする実行コードの前記秘密保護属性保持部と比較することによって同一の暗号鍵か否かを判断することを特徴とする請求項２に記載の耐タンパマイクロプロセッサ。

【請求項４】　暗号化されたプログラムのデータの読み出し要求を処理するキャッシュメモリ制御手段と、

前記キャッシュメモリ制御手段からの復号化要求に基づき、前記データを記憶装置より読み出し、所定の暗号鍵によって復号化する復号化手段と、

前記復号化手段にて復号化した前記データを記憶するキャッシュメモリとを具備し、

前記キャッシュメモリ制御手段は、

前記データの読み出し要求を処理する際に、前記キャッシュメモリ内に前記データが存在し、かつ該データが前記所定の暗号鍵と同一の暗号鍵によって復号化されたデータであった場合には、前記復号化手段へ前記復号化要求を出す代わりに、前記キャッシュメモリ内のデータを使用するよう制御することを特徴とする耐タンパマイクロプロセッサ。

【請求項５】　前記データは、復号化後に前記キャッシュメモリ内のキャッシュライン毎に記憶され、前記キャッシュラインは前記データの復号化に用いた暗号鍵を格納するための秘密保護属性保持部を有することを特徴とする請求項４に記載の耐タンパマイクロプロセッサ。

【請求項６】　前記暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵が更新されて記憶される鍵値レジスタを備え、

前記キャッシュメモリ制御部は、前記キャッシュメモリ内のデータが前記所定の暗号鍵と同一の暗号鍵によって復号化されたデータである否かを判断する際、前記鍵値レジスタと、実行しようとするデータの前記秘密保護属性保持部と比較することによって同一の暗号鍵か否かを判断することを特徴とする請求項５に記載の耐タンパマイクロプロセッサ。

【請求項７】　前記キャッシュメモリ制御部は、前記秘密保護属性保持部内の暗号鍵を用いて前記データの処理結果を暗号化し、前記記憶装置に記憶させることを特徴とする請求項５又は６に記載の耐タンパマイクロプロセッサ。

【請求項８】前記キャッシュメモリ制御部は、前記データの処理結果をキャッシュメモリに書き込む際、前記鍵値レジスタの暗号鍵を前記秘密保護属性保持部に格納することを特徴とする請求項５乃至７のいずれか１項に記載の耐タンパマイクロプロセッサ。

【請求項９】　暗号化されたプログラムの実行コードの読み出し要求を処理

するステップと、

前記実行コードを記憶装置より読み出し、所定の暗号鍵によって復号化するステップと、

復号化された前記実行コードをキャッシュメモリ内に記憶するステップと、

前記実行コードの読み出し要求を処理する際に、前記キャッシュメモリ内に前記実行コードが存在し、かつ該実行コードが前記所定の暗号鍵と同一の暗号鍵によって復号化された実行コードであった場合には、前記キャッシュメモリ内の実行コードを使用するよう制御するステップ

とを備えることを特徴とするキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【請求項１０】　前記キャッシュメモリ内のキャッシュライン毎に前記実行コードの復号化に使用した暗号鍵を秘密保護属性保持部に格納するステップと、

前記暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵を更新し、鍵値レジスタに記憶するステップを備え、

前記キャッシュメモリ内の実行コードを使用するよう制御するステップは、前記鍵値レジスタと、実行しようとする実行コードの前記秘密保護属性保持部と比較することによって同一の暗号鍵か否かを判断することを特徴とする請求項９に記載のキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【請求項１１】　暗号化されたプログラムのデータの読み出し要求を処理するステップと、

前記データを記憶装置より読み出し、所定の暗号鍵によって復号化するステップと、

復号化された前記データをキャッシュメモリ内に記憶するステップと、

前記データの読み出し要求を処理する際に、前記キャッシュメモリ内に前記データが存在し、かつ該データが前記所定の暗号鍵と同一の暗号鍵によって復号化されたデータであった場合には、前記キャッシュメモリ内のデータを使用するよう制御するステップ

とを備えることを特徴とするキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【請求項１２】　前記キャッシュメモリ内のキャッシュライン毎に前記データの復号化に使用した暗号鍵を秘密保護属性保持部に格納するステップと、

前記暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵を更新し、鍵値レジスタに記憶するステップを備え、

前記キャッシュメモリ内のデータを使用するよう制御するステップは、前記鍵値レジスタと、実行しようとするデータの前記秘密保護属性保持部と比較することによって同一の暗号鍵か否かを判断することを特徴とする請求項１１に記載のキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【請求項１３】　前記秘密保護属性保持部内の暗号鍵を用いて前記データの処理結果を暗号化し、前記記憶装置に記憶させるステップを含むことを特徴とする請求項１１又は１２に記載のキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【請求項１４】前記データの処理結果をキャッシュメモリに書き込む際、前記鍵値レジスタの暗号鍵を前記秘密保護属性保持部に格納するステップを含むことを特徴とする請求項１１乃至１３のいずれか１項に記載のキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、マルチタスクのプログラム実行環境を支援する機能を持つマイクロプロセッサであって、キャッシュメモリ制御部及び暗号化・復号化機能を備え、プログラムの実行コード及び処理対象データの秘匿性の保護及び改変を防止する為の耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法に関する。

【０００２】

【従来の技術】

近年のコンピュータシステムにおいては、パーソナルコンピュータのように、様々なメーカーのハードウェア及びソフトウェアの組み合わせ構築が可能なオープンシステムが広く普及ししている。オープンシステムでは、ハードウェア及び

システムプログラム等から成るオペレーティングシステム（以下「ＯＳ」と記載）の情報が一般に開示されている。この為、ユーザが開示情報をもとにＯＳのプログラムを改変・改竄することは原理的に可能である。

【０００３】

アプリケーションプログラムはこのＯＳの管理下で動作するので、ＯＳ自体が改竄されてハッカー等の第三者により攻撃された場合、これを避ける手段がない。よって、アプリケーションプログラム提供者が、アプリケーションプログラムを第三者の解析や改竄から完全に保護することは困難であった。

【０００４】

そこで、オープンシステムのＯＳで動作するアプリケーションプログラムの改竄及び解析を防止する為に、暗号化という手法を用いる。プログラムを暗号化した場合、解析が困難となり、そのプログラムを改変した場合の動作が予測困難となる為改竄の防止にも有効である。ただし暗号化したアプリケーションプログラムを既存のコンピュータで実行することはできないので、これを復号化しつつ実行するようなマイクロプロセッサが必要となる。このマイクロプロセッサは、ＯＳがアプリケーションプログラムに対して敵対動作をとることを前提として、プログラムの秘密を守り、プログラムだけでなくプログラムが扱う情報やデータも暗号化して、解析や改竄から保護する機能を備える（以下、このようなマイクロプロセッサを「耐タンパマイクロプロセッサ」と記載）。又、この耐タンパマイクロプロセッサは保護を受けたプログラムを複数同時に疑似並列実行するマルチタスクのプログラム実行環境を提供する（例えば特許文献１及び非特許文献１参照。）。

【０００５】

耐タンパマイクロプロセッサがキャッシュメモリを搭載する場合、暗号化や復号化を行なう暗号処理部は、プロセッサコアとキャッシュメモリの間若しくはキャッシュメモリとメインメモリ等の記憶装置の間に配置することが可能である。暗号処理部をキャッシュメモリとメインメモリの間に配置すると、キャッシュメモリには復号化後、又は暗号化前の平文の内容が格納される。このため前者よりも後者の配置手法が、暗号化・復号化の処理回数が少なくて済むため、効率的で

ある。

【０００６】

暗号処理部が暗号化や復号化を行なう際、マルチタスクのプログラム実行環境であると、キャッシュメモリ内には複数のプログラムやそれらのデータが格納されている。この時、第三者によるＯＳ等の改竄により、キャッシュメモリ内のプログラム間において、他のプログラムの秘密情報等の盗聴、改竄等が発生することがある。これを防止する為、キャッシュメモリに対するアクセスの制限が必要となる。

【０００７】

従来提案されてきた耐タンパマイクロプロセッサでは、同時に動作する各プログラムに対して１つずつタスクＩＤを与え、これをキャッシュメモリに対するアクセス制限に用いる。キャッシュメモリの各キャッシュラインにはタスクＩＤを格納するためのメモリとして秘密保護フィールドが用意される。プロセッサコアがキャッシュメモリに平文の実行コードやデータを格納するときは、対応する秘密保護フィールドに現在実行中のプログラムのタスクＩＤを格納する。プロセッサコアがキャッシュメモリの内容を読み出す際は、読み出そうとしているキャッシュラインの秘密保護フィールドからタスクＩＤを取得する。そのタスクＩＤと、現在実行中のプログラムのタスクＩＤとを比較し、これらが一致する場合のみ読み出しを許可する。キャッシュメモリに格納されたデータをメインメモリ等の記憶装置に書き出すときには暗号化を行なう必要があるが、その際に用いる暗号鍵は、現在実行中のプログラムが保持する暗号鍵であるとは限らない。従来提案されてきた耐タンパマイクロプロセッサでは、キャッシュラインの秘密保護フィールドに格納されたタスクＩＤにより暗号鍵を取得する。このためタスクＩＤと暗号鍵の対応関係を格納した鍵値テーブルをプロセッサ内部に保持する。又、複数のプログラムが同時に協調動作をするとき、これらのプログラムの間で他のプログラムからは読み出せないようにデータを共有する機能がある。この機能を実現するにはプログラムどうしで１つの暗号鍵の値を共有する。各々のプログラムがこの１つの暗号鍵をデータを読み書きするために使用すれば、これらのプログラムどうしで共有されるメモリ領域の内容を共有することができ、この暗号鍵を

知らない他のプログラムは、同じメモリ領域の内容を正しく暗号化・復号化して読み書きすることができない（例えば特許文献２参照。）。

【０００８】

【特許文献１】

特開２００１－２３０７７０号公開公報

【０００９】

【特許文献２】

特開２００２－２０２７２０号公開公報

【００１０】

【非特許文献１】

David Lie、Chandramohan Thekkath、Mark Mitchell、Patrick Lincoln、Dan Boneh、John Mitchell、Mark Horowitz著「Architectural Support for Copy and Tamper Resistant Software」、ACM 発行「ASPLOS-IX Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA, USA, November 12-15, 2000」、２０００年１１月、ｐ．１６８－１７７

【００１１】

【発明が解決しようとする課題】

しかしながら、上記のタスクＩＤを使用してキャッシュメモリに対するアクセスの制限をする方式には、以下に述べるような問題点が存在する。

【００１２】

先ず、第１の問題点として、この方式の耐タンパマイクロプロセッサはタスクＩＤと暗号鍵の対応関係を格納した鍵値テーブルを保持するが、この鍵値テーブルの大きさにより同時起動可能なプログラムの数が制限される。

【００１３】

第２の問題点として、あるタスクＩＤのプログラムを終了させてそのタスクＩＤを新たに起動する別のプログラムに割り当てるとき、前に実行されていたプログラムのデータがキャッシュメモリに残っていると、同じタスクＩＤを割り当てられた次のプログラムは暗号鍵を知らなくてもそのデータの読み出しが可能とな

る。これを回避するには、プログラム終了時にキャッシュメモリの全領域を走査し、終了したプログラムのタスクＩＤに対応する各キャッシュラインを無効化する必要があるが、この操作には時間が長くかかった。特に大きなキャッシュメモリを搭載するマイクロプロセッサではそれが顕著であった。

【００１４】

第３の問題点として、複数のプログラムが同時に協調動作する際にメモリの内容を共有する場合、あるプログラムが書き込んだデータを別のプログラムが読み出すには、最初のプログラムが書き込んだデータをいったんキャッシュメモリからメインメモリ等の記憶装置に暗号化して書き出し、これを再度復号化してキャッシュメモリに読み込む必要がある。つまり、本来同一内容のデータであるにもかかわらず、それを一度暗号化し、再び同じ鍵で復号化するため、実行速度が低下する。

【００１５】

最後に、第４の問題点として、タスクＩＤを使用してキャッシュメモリに対するアクセスの制限をする方式では鍵値テーブルを参照する機能及びプログラム終了時にキャッシュ領域を走査し無効化する機能等、多くの機能をハードウェアで実装する必要があり、マイクロプロセッサの構造が複雑化する。

【００１６】

本発明はこれらの問題点を解決する為のものであり、キャッシュメモリに対するアクセスの制限方法において、同時起動可能なプログラムの数に制限を与えず、１つのプログラムが終了したときにそのプログラムが使用したキャッシュメモリの内容をキャッシュメモリの全領域を走査することなく他のプログラムから読み出し不可能とし、複数のプログラムが１つの暗号鍵を共有して共有メモリ領域を読み書きする際に、あるプログラムがキャッシュに書き込んだ内容を、それを暗号化して主メモリに書き出さずに他のプログラムから読み出すことができ、複雑なハードウェア機能が不要な耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法を提供することを目的とする。

【００１７】

【課題を解決するための手段】

上記の問題点を鑑み、本発明の第１の特徴は、（イ）暗号化されたプログラムの実行コードの読み出し要求を処理するキャッシュメモリ制御手段と、（ロ）キャッシュメモリ制御手段からの復号化要求に基づき、実行コードを記憶装置より読み出し、所定の暗号鍵によって復号化する復号化手段と、（ハ）復号化手段にて復号化した実行コードを記憶するキャッシュメモリとを具備し、（ニ）キャッシュメモリ制御手段は、実行コードの読み出し要求を処理する際に、キャッシュメモリ内に実行コードが存在し、かつ実行コードが所定の暗号鍵と同一の暗号鍵によって復号化された実行コードであった場合には、復号化手段へ復号化要求を出す代わりに、キャッシュメモリ内の実行コードを使用するよう制御する耐タンパマイクロプロセッサであることを要旨とする。又、本発明の第１の特徴は、（ホ）実行コードは、復号化後にキャッシュメモリ内のキャッシュライン毎に記憶され、キャッシュラインは実行コードの復号化に用いた暗号鍵を格納するための秘密保護属性保持部を有すること、（ヘ）暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵が更新されて記憶される鍵値レジスタを備え、キャッシュメモリ制御部は、キャッシュメモリ内の実行コードが所定の暗号鍵と同一の暗号鍵によって復号化された実行コードである否かを判断する際、鍵値レジスタと、実行しようとする実行コードの秘密保護属性保持部と比較することによって同一の暗号鍵か否かを判断することを加えても良い。

【００１８】

上記の発明によると、キャッシュメモリに対するアクセス制限が無い為、同時起動可能なプログラムの数に制限がない。又、１つのプログラムが終了したときにそのプログラムが使用していたキャッシュメモリの内容は、そのプログラムの暗号鍵を知らない限り読み出すことができない。その為、キャッシュメモリを無効化するためにキャッシュメモリ全領域を走査する必要が無い。さらに、アクセス制限を行う機能や、キャッシュメモリ全領域を走査する機能が不要であるため、ハードウェアの複雑化を防ぐことができる。

【００１９】

本発明の第２の特徴は、（イ）暗号化されたプログラムのデータの読み出し要求を処理するキャッシュメモリ制御手段と、（ロ）キャッシュメモリ制御手段か

らの復号化要求に基づき、データを記憶装置より読み出し、所定の暗号鍵によって復号化する復号化手段と、（ハ）復号化手段にて復号化したデータを記憶するキャッシュメモリとを具備し、（ニ）キャッシュメモリ制御手段は、データの読み出し要求を処理する際に、キャッシュメモリ内にデータが存在し、かつデータが所定の暗号鍵と同一の暗号鍵によって復号化されたデータであった場合には、復号化手段へ復号化要求を出す代わりに、キャッシュメモリ内のデータを使用するよう制御する耐タンパマイクロプロセッサであることを要旨とする。

【００２０】

本発明の第３の特徴は、（イ）暗号化されたプログラムの実行コードの読み出し要求を処理するステップと、（ロ）実行コードを記憶装置より読み出し、所定の暗号鍵によって復号化するステップと、（ハ）復号化された実行コードをキャッシュメモリ内に記憶するステップと、（ニ）実行コードの読み出し要求を処理する際に、キャッシュメモリ内に実行コードが存在し、かつ実行コードが所定の暗号鍵と同一の暗号鍵によって復号化された実行コードであった場合には、キャッシュメモリ内の実行コードを使用するよう制御するステップとを備えるキャッシュメモリ搭載プロセッサによるデータアクセス制御方法であることを要旨とする。又、本発明の第３の特徴は、（ホ）キャッシュメモリ内のキャッシュライン毎に実行コードの復号化に使用した暗号鍵を秘密保護属性保持部に格納するステップと、（ヘ）暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵を更新し、鍵値レジスタに記憶するステップを備え、（ト）キャッシュメモリ内の実行コードを使用するよう制御するステップは、鍵値レジスタと、実行しようとする実行コードの秘密保護属性保持部と比較することによって同一の暗号鍵か否かを判断することを加えても良い。

【００２１】

本発明の第４の特徴は、（イ）暗号化されたプログラムのデータの読み出し要求を処理するステップと、（ロ）データを記憶装置より読み出し、所定の暗号鍵によって復号化するステップと、（ハ）復号化されたデータをキャッシュメモリ内に記憶するステップと、（ニ）データの読み出し要求を処理する際に、キャッシュメモリ内にデータが存在し、かつデータが所定の暗号鍵と同一の暗号鍵によ

って復号化されたデータであった場合には、キャッシュメモリ内のデータを使用するよう制御するステップとを備えるキャッシュメモリ搭載プロセッサによるデータアクセス制御方法であることを要旨とする。

【００２２】

【発明の実施の形態】

キャッシュメモリ制御部及び暗号化復号化機能を備え、プログラムの実行コード及び処理対象データの秘匿性の保護及び改変を防止する為の耐タンパマイクロプロセッサ及びデータアクセス制御方法の実施の形態について以下に説明する。尚、本発明は以下の実施の形態にとらわれず、その趣旨を脱しない範囲で種々変更して実施できることは勿論である。

【００２３】

（耐タンパマイクロプロセッサ）

本発明の実施の形態に係る耐タンパマイクロプロセッサ１００は、図１に示すように、プロセッサコア１０、キャッシュメモリ制御部２０、コードデータ暗号復号処理部３０、鍵値レジスタ４０及び外部バスインタフェース５０等を備える。耐タンパマイクロプロセッサ１００は、外部バスインタフェース５０に接続されたバス７０を介して記憶装置６０と接続されている。

【００２４】

プロセッサコア１０は、キャッシュメモリ制御部２０から渡された平分コード及び平文データの演算処理を行う。

【００２５】

コードデータ暗号復号処理部３０は、復号化手段であり、キャッシュメモリ制御部２０と外部バスインタフェース５０の間に配置され、バス７０側から送信された暗号化済み実行コードやデータの復号を行うモジュールである。またコードデータ暗号復号処理部３０は、キャッシュメモリ制御部２０から受信したデータを暗号化しバス７０側に出力する。

【００２６】

鍵値レジスタ４０は、コードデータ暗号復号処理部３０が暗号化及び復号化の実行時に使う暗号鍵を格納する。１つのプログラムは図５のように、複数の実行

コード及びデータから構成され、これら複数の実行コード及びデータは、鍵値レジスタ４０内の１つの暗号鍵によって暗号化若しくは復号化される。なお、本実施の形態ではプログラムの持つ実行コード及びデータは同一の暗号鍵で暗号化されるものとするが、実行コードとデータを別々の鍵で暗号化するために、鍵値レジスタ４０に実行コード用とデータ用の２つの暗号鍵を保持できるようにしても良い。鍵値レジスタ４０と図２の秘密保護属性保持部２５ａ～２５ｄは同じ鍵値を格納する為、これらの領域と同じ大きさであることが好ましい。鍵値レジスタ４０の領域は、新たなプログラムを実行する際には新たなプログラムの鍵値に上書きされ、更新される。

【００２７】

記憶装置６０は、プログラムのコンパイル後の実行コード及びデータを記憶する為の主メモリ、記憶装置等を指す。又記憶装置６０は、バス７０を介して図示しない通信制御装置を用い、外部の補助記憶装置若しくは外部バスとのデータのやり取りを行っても良い。尚、記憶装置６０が記憶する実行コード及びデータは既に相手の暗号処理装置等によって暗号化された状態で受信したものを記憶している。

【００２８】

キャッシュメモリ制御部２０は、キャッシュメモリ制御手段であり、実行コード及びデータに付随するアドレス情報や秘密保護属性等の情報の比較処理、演算処理等の制御を行う。キャッシュメモリ制御部２０は、キャッシュメモリ２１を内部に備え、キャッシュメモリ２１は実行コード及びデータを平文コード及び平文データとして記憶する。

【００２９】

図２は、キャッシュメモリ２１のデータ構造を示す。キャッシュメモリ２１は複数のキャッシュライン２２ａ、２２ｂ、２２ｃ、２２ｄ…（以下「キャッシュライン２２ａ～２２ｄ」と記載）から構成される。キャッシュライン２２ａ～２２ｄは、タグ領域２３ａ、２３ｂ、２３ｃ、２３ｄ…（以下「タグ領域２３ａ～２３ｄ」と記載）、データ領域２４ａ、２４ｂ、２４ｃ、２４ｄ…（以下「データ領域２４ａ～２４ｄ」と記載）、秘密保護属性保持部２５ａ、２５ｂ、２５ｃ

、２５ｄ…（以下「秘密保護属性保持部２５ａ～２５ｄ」と記載）及び制御情報保持部２６ａ、２６ｂ、２６ｃ、２６ｄ…（以下「制御情報保持部２６ａ～２６ｄ」と記載）等から構成される。タグ領域２３ａ～２３ｄは、実行コード及びデータが読み出されたアドレスに関する情報を記憶する領域である。データ領域２４ａ～２４ｄは、読み出された実行コード及びデータを記憶する領域である。秘密保護属性保持部２５ａ～２５ｄは、実行コード及びデータの復号化に使用した暗号鍵、データの暗号化に使用するべき暗号鍵を記憶する領域である。制御情報保持部２６ａ～２６ｄは、各キャッシュライン２２ａ～２２ｄが保持する実行コード及びデータの制御情報を記憶する領域である。

【００３０】

本発明の実施の形態に係る耐タンパマイクロプロセッサが、鍵値レジスタの値の初期設定、実行コードの読み出し、データの読み出し及びデータの書き出しを行う際の動作について以下に説明する。

【００３１】

（鍵値レジスタの値の初期設定）

記憶装置６０にロードされたプログラムは、暗号化された実行コードによる実行を開始する前に、プログラム実行の初期設定として、暗号鍵を耐タンパマイクロプロセッサ１００の鍵値レジスタ４０に設定する。具体的には、暗号鍵を符合化したデータをパラメータとしてプロセッサコア１０に与える。プロセッサコア１０は、与えられた暗号鍵を鍵値レジスタ４０に設定する。例えば、図５のように、プログラム１において実行コード若しくはデータ「Ａ」「Ｂ」「Ｃ」を実行する前に、鍵値「Ｘ」を鍵値レジスタ４０に設定し、プログラム２において実行コード若しくはデータ「Ｄ」「Ｅ」「Ｆ」を実行する前に、鍵値「Ｙ」を鍵値レジスタ４０に設定する。又は、鍵値が０の、暗号化されていない実行コードを、図５中の「Ａ」の前に挿入し、その実行コードを実行する際の動作内容として鍵値レジスタ４０を設定してもよい。

【００３２】

尚、暗号化されたプログラムの実行中に、割り込み等により実行が中断されて他のプログラムに制御が移る場合には、通常のコンテキスト保存操作の一部とし

て鍵値レジスタの値を保存する。また、中断されていたプログラムの実行を再開する際も、通常のコンテキスト復元操作の一部として鍵値レジスタの値を復元する。ここでコンテキストの暗号化を行なっても良い。鍵値レジスタをコンテキストとして保存することにより、別々の暗号鍵で暗号化された複数のプログラムが疑似並列的に実行されるマルチタスク環境が実現される。

【００３３】

耐タンパマイクロプロセッサ１００の電源投入時やリセットの直後、及びコンテキスト保存の直後（ただしコンテキスト保存の直後に別のコンテキストの復元が行なわれる場合は除く）の制御として、耐タンパマイクロプロセッサ１００は鍵値レジスタ４０に特別な値の設定する。これは例えば、値がゼロの暗号鍵、鍵値レジスタの１ビットをフラグとする等の手法がある。鍵値レジスタ４０にこれらの特別な値が設定されているとき、耐タンパマイクロプロセッサ１００は記憶装置６０等から読み出した実行コードの復号化を行なわず、読み出したものをそのまま実行する。このことによりプログラマ等は暗号化復号化を行うプログラムや、暗号化復号化を行うタイミング等を自由に設定することが可能となる。

【００３４】

プロセッサコア１０により鍵値レジスタ４０に暗号鍵が設定された後、キャッシュメモリ制御部２０は実行コード及びデータの読み出しを以下で述べる手順により実行する。

【００３５】

（実行コード及びデータの読み出し）

本発明の実施の形態に係る耐タンパマイクロプロセッサ１００が、暗号化されたプログラムの実行コード又はデータを記憶装置６０又はキャッシュメモリ２１より読み出す動作について、図３のフローチャートを参照し、以下に説明する。

【００３６】

（ａ）先ず、キャッシュメモリ制御部２０は、ステップＳ１０１において、プロセッサコア１０より実行コード又はデータ取得の要求を受け取ると、ステップＳ１０２にて、第１の確認として、この実行コード又はデータがキャッシュメモリ２１上にあるかを判断する。

【００３７】

（ｂ）実行すべきアドレスの実行コード又はデータがキャッシュメモリ２１上にないと判断された場合は、ステップＳ１０３にて、バス７０及び外部バスインタフェース５０を経由して記憶装置６０からそのアドレスの実行コード又はデータを読み出す。

【００３８】

（ｃ）ステップＳ１０４では、コードデータ暗号復号処理部３０が、記憶装置６０より読み出された実行コード又はデータを、初期設定された鍵値レジスタ４０の暗号鍵で復号化する。

【００３９】

（ｄ）ステップＳ１０５にて、この実行コード又はデータはキャッシュメモリ２１に転送され、図２のキャッシュライン２２ａ～２２ｄ上のデータ領域２４ａ～２４ｄのいずれかに格納される。この際、ステップＳ１０６にて、実行コード又はデータが読み出されたアドレスに関する情報がタグ領域２３ａ～２３ｄのいずれかに格納され、制御情報保持部２６ａ～２６ｄのいずれかの内容が更新される。また、実行コード又はデータの復号化に用いた鍵値レジスタ４０の暗号鍵は秘密保護属性保持部２５ａ～２５ｄのいずれかに格納される。ステップＳ１０７にて、復号化された実行コード又はデータは実行のためプロセッサコアにも転送される。

【００４０】

（ｅ）一方、ステップＳ１０２において、第１の確認の際、実行すべきアドレスの実行コード又はデータがキャッシュメモリに存在すると判断された場合は、ステップＳ１０８において、キャッシュメモリ制御部２０が、第２の確認を行う。即ち、キャッシュライン２２ａ～２２ｄに含まれる秘密保護属性保持部２５ａ～２５ｄの内、これから実行すべきアドレスの実行コード又はデータのキャッシュライン中の秘密保護属性保持部より暗号鍵を取得する。更に、ステップＳ１０９において、現在演算処理が実行されているプログラムが固有する暗号鍵を格納する鍵値レジスタ４０より暗号鍵を取得する。ステップＳ１１０にて、これらの暗号鍵を比較する。もし２つの暗号鍵が一致した場合、これから実行すべきアドレ

スの実行コード又はデータとして、キャッシュメモリ２１内に既に存在する同一の実行コード又はデータの内容を使用することを許可される。その後ステップＳ１１１にて、キャッシュライン２２ａ～２２ｄのデータ領域２４ａ～２４ｄにある実行コード又はデータ内に存在する同一の実行コード又はデータが再びプロセッサコア１０に転送される。反対にステップＳ１１０において、２つの暗号鍵が一致しない場合は、現在実行中のプログラムはキャッシュメモリの内容を使用することを許可されず、ステップＳ１０３～Ｓ１０７の、実行すべきアドレスの実行コード又はデータがキャッシュメモリにない場合と同様の動作を行なう。又は、プログラムの実行を中断して異常終了を示す例外を発生しても良い。

【００４１】

上記の耐タンパマイクロプロセッサ１００の動作の具体例として、図５のプログラム１及びプログラム２を耐タンパマイクロプロセッサ１００が実行する動作について説明する。プログラム１及びプログラム２を実行する際には、プログラム１の終了後にプログラム２を実行する場合（実施例１）、プログラム１の実行中にプログラム２が割り込んで実行する場合（実施例２）、同じプログラム内で実行コード若しくはデータを実行する場合（実施例３）、異なるプログラム間にて実行コード若しくはデータを実行する場合（実施例４）とが考えられる。以下、それぞれの実施例について詳述する。

【００４２】

（実施例１）

先ず、通常の処理として、図５のプログラム１の終了後にプログラム２を実行する場合について図６を用いて説明する。

【００４３】

（ａ）先ず、耐タンパマイクロプロセッサ１００は、ステップＳ３０１において、プログラム１の初期設定として、鍵値レジスタ４０に「鍵値Ｘ」を格納する。この後、ステップＳ３０２において、記憶装置６０よりプログラム１の実行コード若しくはデータ「Ａ」を呼び出し、「鍵値Ｘ」にて復号化し、図２のキャッシュライン２２ａに格納する。この際、秘密保護属性保持部２５ａには「鍵値Ｘ」が格納される。又、実行コード若しくはデータ「Ａ」はプロセッサコア１０にて

演算処理される。プロセッサコア１０は図示しないプログラムカウンタを１つずつ進め、これと同様の処理を実行コード若しくはデータ「Ｂ」「Ｃ」に対しても行う（ステップＳ３０３～Ｓ３０４）。

【００４４】

（ｂ）プログラム１が完了すると、待機していたプログラム２の処理に入る。ステップＳ３０５において、プログラム２の初期設定として、鍵値レジスタ４０に「鍵値Ｙ」を格納する。この後、ステップＳ３０６において、記憶装置６０よりプログラム２の実行コード若しくはデータ「Ｄ」を呼び出し、「鍵値Ｙ」にて復号化し、図２のキャッシュライン２２ｄに格納する。この際、秘密保護属性保持部２５ｄには「鍵値Ｙ」が格納される。又、実行コード若しくはデータ「Ｄ」はプロセッサコア１０にて演算処理される。プロセッサコア１０は図示しないプログラムカウンタを１つずつ進め、これと同様の処理を実行コード若しくはデータ「Ｅ」「Ｆ」に対しても行う（ステップＳ３０７～Ｓ３０８）。

【００４５】

実施例１によると、暗号化されたプログラム１及びプログラム２を、暗号鍵を用いて平文化することにより、悪意の第三者の盗聴及び改竄等より防止し、安全に実行することができる。

【００４６】

（実施例２）

次に、図５のプログラム１の実行中にプログラム２が割り込んで実行する場合について図７を用いて説明する。

【００４７】

（ａ）先ず、耐タンパマイクロプロセッサ１００は、ステップＳ４０１において、プログラム１の初期設定として、鍵値レジスタ４０に「鍵値Ｘ」を格納する。この後、ステップＳ４０２において、記憶装置６０よりプログラム１の実行コード若しくはデータ「Ａ」を呼び出し、「鍵値Ｘ」にて復号化し、図２のキャッシュライン２２ａに格納する。この際、秘密保護属性保持部２５ａには「鍵値Ｘ」が格納される。又、実行コード若しくはデータ「Ａ」はプロセッサコア１０にて演算処理される。プロセッサコア１０は図示しないプログラムカウンタを１つず

つ進め、これと同様の処理を実行コード若しくはデータ「Ｂ」に対しても行う（ステップＳ４０３）。

【００４８】

（ｂ）実行コード若しくはデータ「Ｂ」の処理最中若しくは終了後に、プログラム２の割り込みが発生する。ステップＳ４０４において、キャッシュメモリ制御部２０は、通常のコンテキスト保存として「鍵値Ｘ」を保存する。

【００４９】

（ｃ）割り込みしたプログラム２は、ステップＳ４０５において、初期設定として、鍵値レジスタ４０に「鍵値Ｙ」を格納する。この後、ステップＳ４０６において、記憶装置６０よりプログラム２の実行コード若しくはデータ「Ｄ」を呼び出し、「鍵値Ｙ」にて復号化し、図２のキャッシュライン２２ｄに格納する。この際、秘密保護属性保持部２５ｄには「鍵値Ｙ」が格納される。又、実行コード若しくはデータ「Ｄ」はプロセッサコア１０にて演算処理される。プロセッサコア１０は図示しないプログラムカウンタを１つずつ進め、これと同様の処理を実行コード若しくはデータ「Ｅ」「Ｆ」に対しても行う（ステップＳ４０７～Ｓ４０８）。

【００５０】

（ｄ）プログラム２が終了すると、割り込みされたプログラム１は処理を再開する。ステップＳ４０９として、保存されていた「鍵値Ｘ」を鍵値レジスタ４０に再設定し、一時中断されていたプログラム１の実行コード若しくはデータ「Ｃ」の処理を再開する。

【００５１】

実施例２によると、暗号化されたプログラム１及びプログラム２を復号するための暗号鍵をコンテキストとして保存することにより、複数のプログラムが擬似並列的に実行されるマルチタスク環境を実現することができる。

【００５２】

（実施例３）

次に、図５のプログラム１内で実行コード若しくはデータ「Ａ」「Ｂ」「Ｃ」「Ａ」を実行する場合について図８を用いて説明する。尚、変数ＣＮＴの初期値

は１とする。

【００５３】

（ａ）先ず、耐タンパマイクロプロセッサ１００は、ステップＳ５０１において、プログラム１の初期設定として、鍵値レジスタ４０に「鍵値Ｘ」を格納する。この後、ステップＳ５０２において、記憶装置６０よりプログラム１の実行コード若しくはデータ「Ａ」を呼び出し、「鍵値Ｘ」にて復号化し、図２のキャッシュライン２２ａに格納する。この際、秘密保護属性保持部２５ａには「鍵値Ｘ」が格納される。又、実行コード若しくはデータ「Ａ」はプロセッサコア１０にて演算処理される。プロセッサコア１０は図示しないプログラムカウンタを１つずつ進め、これと同様の処理を実行コード若しくはデータ「Ｂ」「Ｃ」に対しても行う（ステップＳ５０３～Ｓ５０５）。

【００５４】

（ｂ）次に、ステップＳ５０６においてカウンタが１つ進められ、ステップＳ５０７として、再度Ｓ５０２の処理が行われる。この時には、鍵値レジスタ４０は「鍵値Ｘ」を格納しており、実行コード若しくはデータ「Ａ」の秘密保護属性保持部２５ａの値も「鍵値Ｘ」であるため、キャッシュメモリ制御部２０は、キャッシュメモリ２１に存在する実行コード若しくはデータ「Ａ」の使用を許可し、プロセッサコア１０へ対しキャッシュメモリ２１に存在する実行コード若しくはデータ「Ａ」を再送信する。

【００５５】

（ｃ）プログラム１が完了すると、ステップＳ５０８の分岐命令によって、待機していたプログラム２の処理に入る。プログラム２の処理ステップＳ５０９～Ｓ５１２についてはステップＳ３０５～Ｓ３０８と同様である為説明を省略する。

【００５６】

実施例３によると、同一暗号鍵を有するプログラム、つまり同一プログラムにおいては、キャッシュメモリ内に既存する実行コード及びメモリを使用することができる。よって、演算処理効率が向上する。

【００５７】

（実施例４）

最後に、図５のプログラム１及びプログラム２間で実行コード若しくはデータを実行する場合について図９を用いて説明する。

【００５８】

（ａ）先ず、耐タンパマイクロプロセッサ１００は、プログラム１の初期設定と実行コード又はデータ「Ａ」「Ｂ」「Ｃ」に関する処理をステップＳ６０１～Ｓ６０４にて行う。この処理はステップＳ３０１～Ｓ３０４と同様である為説明を省略する。

【００５９】

（ｂ）プログラム１が完了すると、待機していたプログラム２の処理に入る。プログラム２は、ステップＳ６０５において、初期設定として、鍵値レジスタ４０に「鍵値Ｙ」を格納する。この後、ステップＳ６０６において、記憶装置６０よりプログラム２の実行コード若しくはデータ「Ｄ」を呼び出し、「鍵値Ｙ」にて復号化し、図２のキャッシュライン２２ｄに格納する。この際、秘密保護属性保持部２５ｄには「鍵値Ｙ」が格納される。又、実行コード若しくはデータ「Ｄ」はプロセッサコア１０にて演算処理される。プロセッサコア１０は図示しないプログラムカウンタを１つずつ進める。

【００６０】

（ｃ）ステップＳ６０７において、次に処理を行う実行コード若しくはデータ「Ｅ」が、JUMP命令、GOTO命令等で実行コード若しくはデータ「Ｃ」に関わるものであるとする。この場合、キャッシュメモリ制御部２０は、キャッシュメモリ２１内に実行コード若しくはデータ「Ｃ」が存在するか否かを判断する。プログラム１の直後にプログラム２を実行した為、キャッシュメモリ２１内の、例えば、キャッシュライン２２ｃには実行コード若しくはデータ「Ｃ」が存在するとする。次にキャッシュメモリ制御部２０は、鍵値レジスタ４０の現在の値「鍵値Ｘ」と、秘密保護属性保持部２５ｃに格納されている「鍵値Ｙ」とを比較する。この場合、鍵値は一致しない為、ステップＳ６０８にて、キャッシュメモリ制御部２０は、キャッシュメモリ２１内のキャッシュライン２２ｃを使用することを許可しない。この後、実行コード若しくはデータ「Ｃ」をステップＳ１０３～１０７の手順により取得したり、プログラム２の実行を中断し、異常終了を示したりす

る。

【００６１】

実施例４においては、異なる暗号鍵を有するプログラム間、つまり、複数の別々のプログラム間においては、キャッシュメモリ内の実行コード及びデータの使用を許可しない。このことによって、１つのプログラムの終了時にキャッシュメモリ全領域を走査する必要が無くなり、ひいては、演算処理効率が向上する。

【００６２】

（データの書き出し）

本発明の実施の形態に係る耐タンパマイクロプロセッサ１００がプログラムの要求により、データをキャッシュメモリ２１及び記憶装置６０に書き出す動作について、図４のフローチャートを参照し、以下に説明する。

【００６３】

（ａ）先ずステップＳ２０１では、キャッシュメモリ制御部２０が、プロセッサコア１０より、書き出すべきデータを平文のままキャッシュメモリ２１内に取得し、図２のキャッシュライン２２ａ～２２ｄのデータ領域２４ａ～２４ｄの該当領域に格納する。

【００６４】

（ｂ）この際、ステップＳ２０２では、データを書き込むアドレスに関する情報をタグ領域２３ａ～２３ｄの該当領域に格納し、制御情報保持部２６ａ～２６ｄの該当領域の内容を更新する。また、このデータを記憶装置６０に書き出す際に用いるべき暗号鍵が鍵値レジスタ４０より読み出され、秘密保護属性保持部２５ａ～２５ｄの該当領域に格納される。

【００６５】

（ｃ）次に、データをキャッシュメモリ２１から記憶装置６０に書き出す。この動作は、キャッシュメモリ２１の該当するキャッシュライン２２ａ～２２ｄに別のデータを格納する場合等に必要となる。詳細を説明すると、ステップＳ２０３で、キャッシュメモリ制御部２０がキャッシュライン２２ａ～２２ｄのデータ領域２４ａ～２４ｄの該当領域より平文データを取得する。又、ステップＳ２０４で、秘密保護属性保持部２５ａ～２５ｄの該当領域より暗号鍵を取得する。これ

らの平文データ及び暗号鍵はコードデータ暗号復号処理部３０に転送される。

【００６６】

（ｄ）ステップＳ２０５では、コードデータ暗号復号処理部３０が暗号鍵を用いて、平文データを暗号化する。ステップＳ２０６で、暗号化済みのデータは外部バスインタフェース５０及びバス７０を経由して記憶装置６０に書き出される。

【００６７】

上記によると、耐タンパマイクロプロセッサ１００において演算処理され、変形されたデータは、秘密保護属性保持部の暗号鍵を用いて暗号化される。この為、悪意の第三者に盗聴、改竄されること無く、安全な状態で記憶装置に渡され、データを所望する装置、ネットワーク等に送信することができる。

【００６８】

本発明の実施の形態に係る耐タンパマイクロプロセッサによると、キャッシュメモリに対するアクセスの際に鍵値テーブルが不要であるため、第１の問題点である、同時起動可能なプログラムの数に制限を与えることを回避することができる。

【００６９】

１つのプログラムが終了したときにそのプログラムが使用していたキャッシュメモリは、そのプログラムが保持していた暗号鍵の値を知らない限り読み出すことができない。そのため、このキャッシュメモリの内容を残したまま他のプログラムを起動しても、その内容が他のプログラムに読み出される恐れはなく、キャッシュメモリを無効化するために全領域を走査しなくても良い。よって、第２の問題点である全領域走査時間を短縮し、ひいては、プログラムの実行速度を上げることができる。

【００７０】

複数のプログラムが１つの暗号鍵を共有している場合、どのプログラムも前記共有された暗号鍵を用いてキャッシュメモリの内容を参照することができ、１つのプログラムが書き込んだ内容を別のプログラムが読み出すために、その内容をいったん暗号化しメインメモリ等の記憶装置に書き出す必要はない。この為、第３の問題点である、暗号化復号化によるプログラムの実行速度の低下を防ぐこと

ができる。

【００７１】

この耐タンパマイクロプロセッサを用いた手法によると、鍵値テーブルを用いる手法と異なり、鍵値テーブルを参照する機能や、プログラム終了時にキャッシュ領域を走査し無効化する機能は不要であり、第４の問題点であるハードウェアの複雑化を回避することができ、ひいては製作コストを下げることができる。

【００７２】

【発明の効果】

本発明によると、同時起動可能なプログラムの数に制限を与えず、１つのプログラムが終了したときにそのプログラムが使用したキャッシュメモリの内容をキャッシュメモリの全領域を走査することなく他のプログラムから読み出し不可能とし、複数のプログラムが１つの暗号鍵を共有して共有メモリ領域を読み書きする際に、あるプログラムがキャッシュに書き込んだ内容を、それを暗号化して主メモリに書き出さずに他のプログラムから読み出すことができ、複雑なハードウェア機能が不要な耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法を提供することができる。

【図面の簡単な説明】

【図１】

本発明の実施の形態に係る耐タンパマイクロプロセッサの構成図である。

【図２】

本発明の実施の形態に係る耐タンパマイクロプロセッサのキャッシュメモリのデータ構成を示す図である。

【図３】

本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの手順を示すフローチャートである。

【図４】

本発明の実施の形態に係る耐タンパマイクロプロセッサのデータの書き出しの手順を示すフローチャートである。

【図５】

本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータ構造の概略を示す図である。

【図６】

本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの実施例１を示すフロー図である。

【図７】

本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの実施例２を示すフロー図である。

【図８】

本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの実施例３を示すフロー図である。

【図９】

本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの実施例４を示すフロー図である。

【符号の説明】

１０…プロセッサコア

２０…キャッシュメモリ制御部

２１…キャッシュメモリ

２２ａ～２２ｄ…キャッシュライン

２３ａ～２３ｄ…タグ領域

２４ａ～２４ｄ…データ領域

２５ａ～２５ｄ…秘密保護属性保持部

２６ａ～２６ｄ…制御情報保持部

３０…コードデータ暗号復号処理部

４０…鍵値レジスタ

５０…外部バスインタフェース

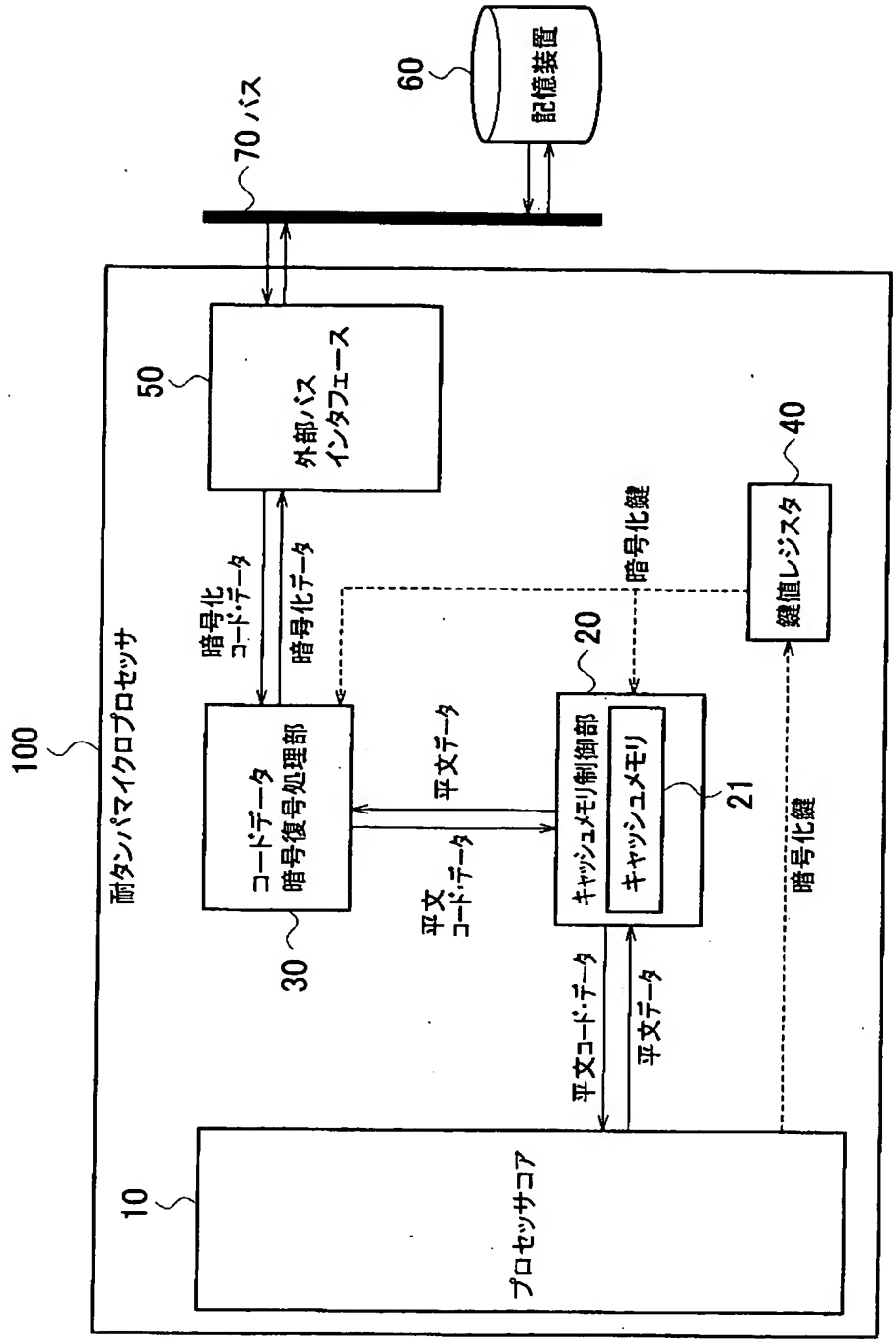
６０…記憶装置

７０…バス

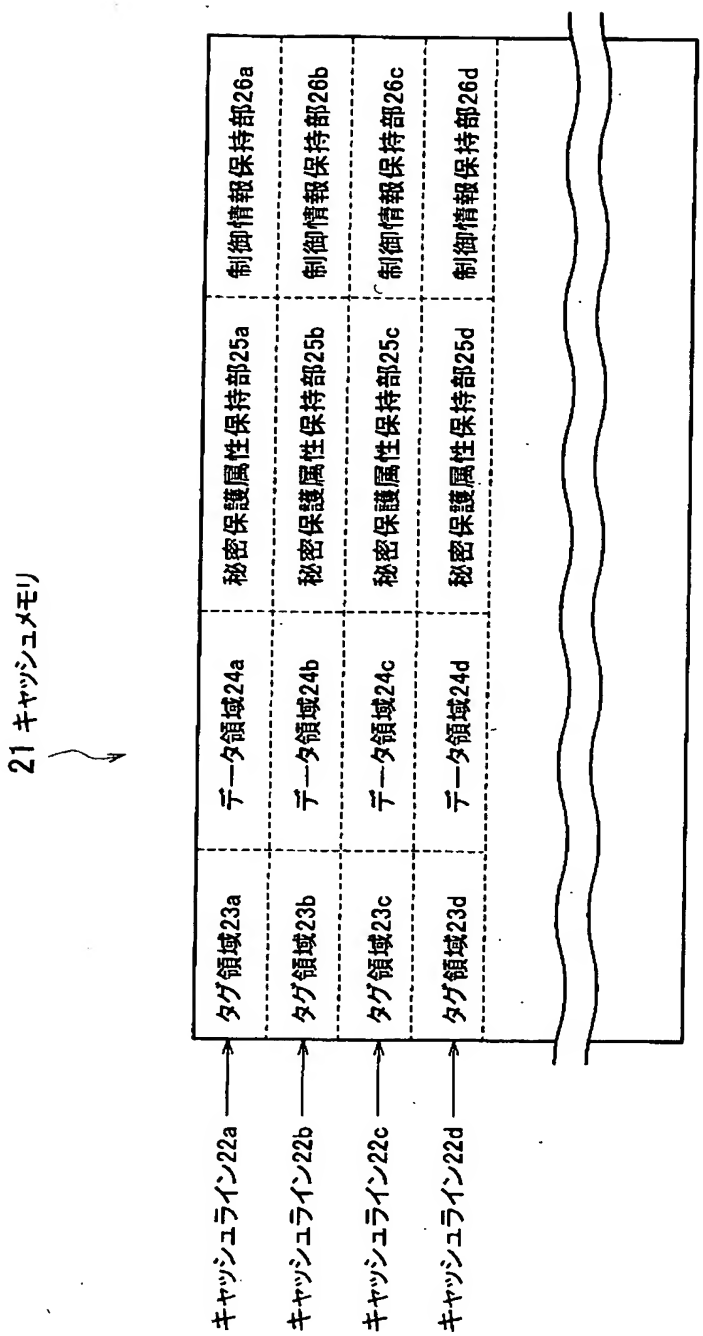
１００…耐タンパマイクロプロセッサ

【書類名】　図面

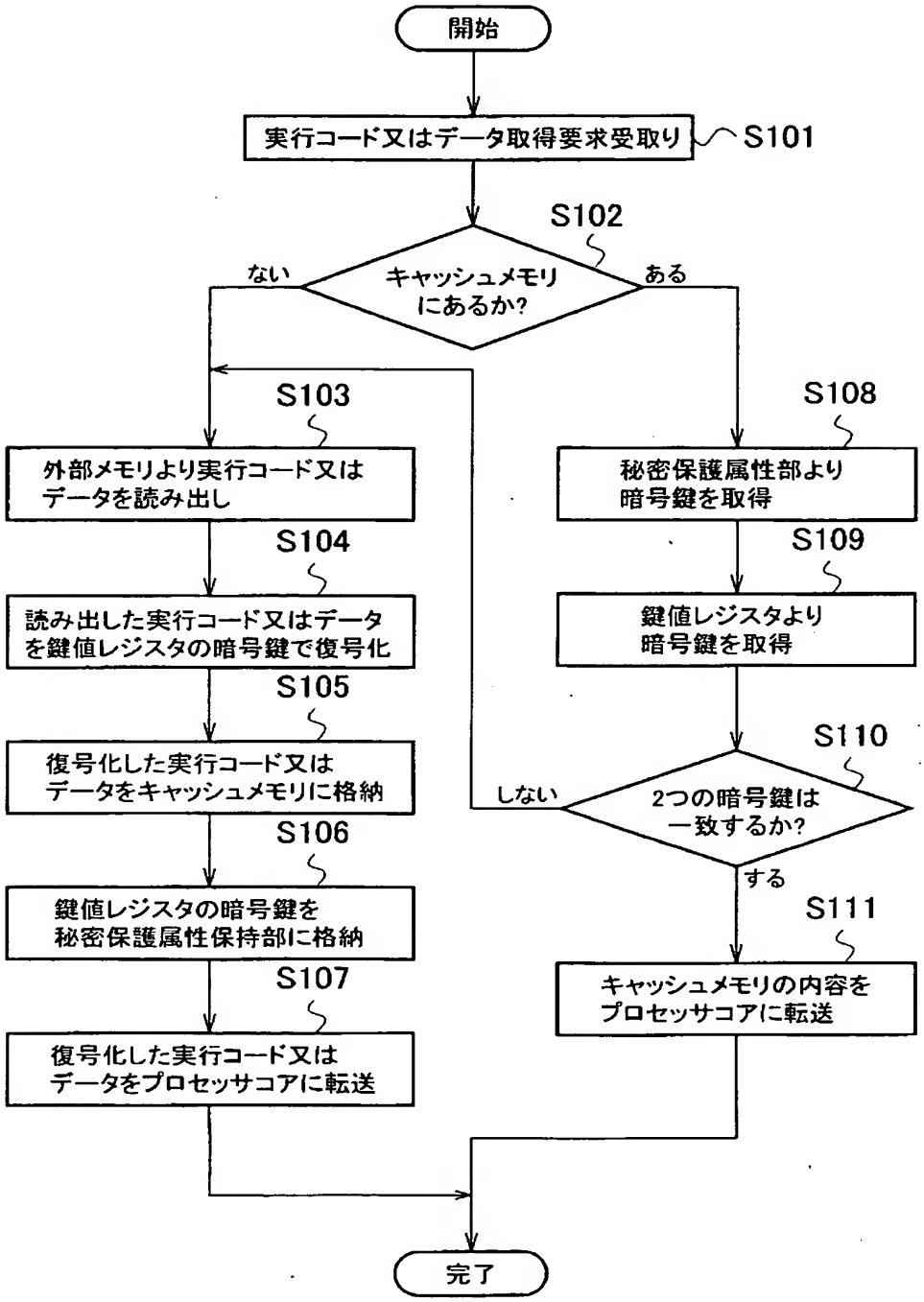
【図１】


60
記憶装置
70 バス
50
外部バス
インタフェース
40
鍵値レジスタ
暗号化
コード・データ
暗号化データ
暗号化鍵
100
耐タンパマイクロプロセッサ
20
コードデータ
暗号復号処理部
平文データ
キャッシュメモリ制御部
キャッシュメモリ
21
平文
コード・データ
暗号化鍵
30
平文コード・データ
平文データ
10
プロセッサコア

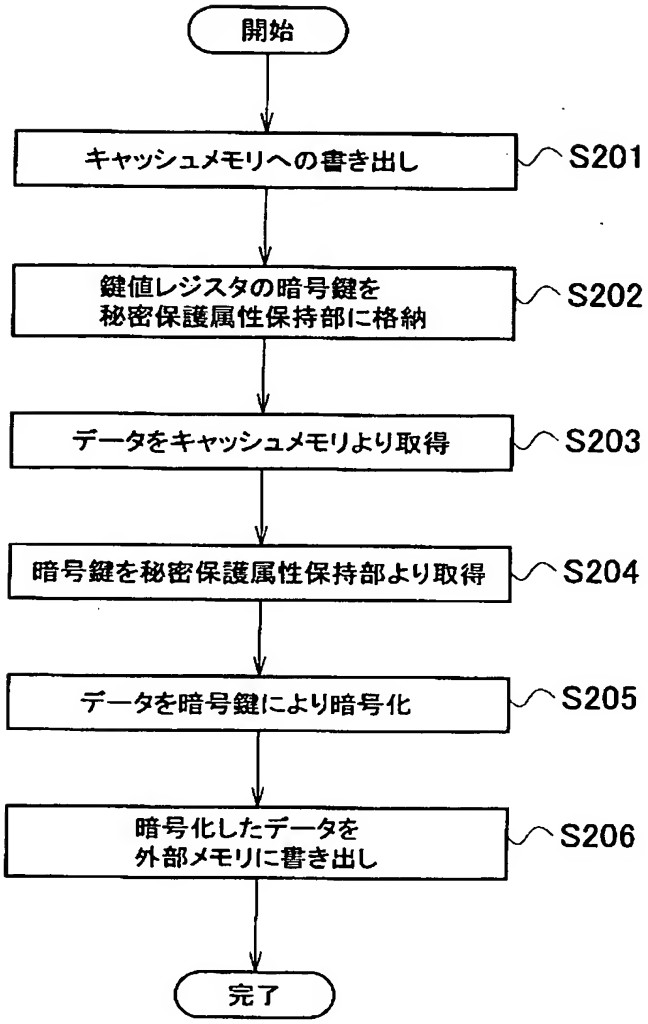
【図２】


21 キャッシュメモリ
タグ領域23a
タグ領域23b
タグ領域23c
タグ領域23d
データ領域24a
データ領域24b
データ領域24c
データ領域24d
秘密保護属性保持部25a
秘密保護属性保持部25b
秘密保護属性保持部25c
秘密保護属性保持部25d
制御情報保持部26a
制御情報保持部26b
制御情報保持部26c
制御情報保持部26d
キャッシュライン22a
キャッシュライン22b
キャッシュライン22c
キャッシュライン22d

【図３】


開始
実行コード又はデータ取得要求受取り
S101
S102
キャッシュメモリにあるか?
ない
ある
S103
外部メモリより実行コード又はデータを読み出し
S104
読み出した実行コード又はデータを鍵値レジスタの暗号鍵で復号化
S105
復号化した実行コード又はデータをキャッシュメモリに格納
S106
鍵値レジスタの暗号鍵を秘密保護属性保持部に格納
S107
復号化した実行コード又はデータをプロセッサコアに転送
S108
秘密保護属性部より暗号鍵を取得
S109
鍵値レジスタより暗号鍵を取得
S110
2つの暗号鍵は一致するか?
しない
する
S111
キャッシュメモリの内容をプロセッサコアに転送
完了

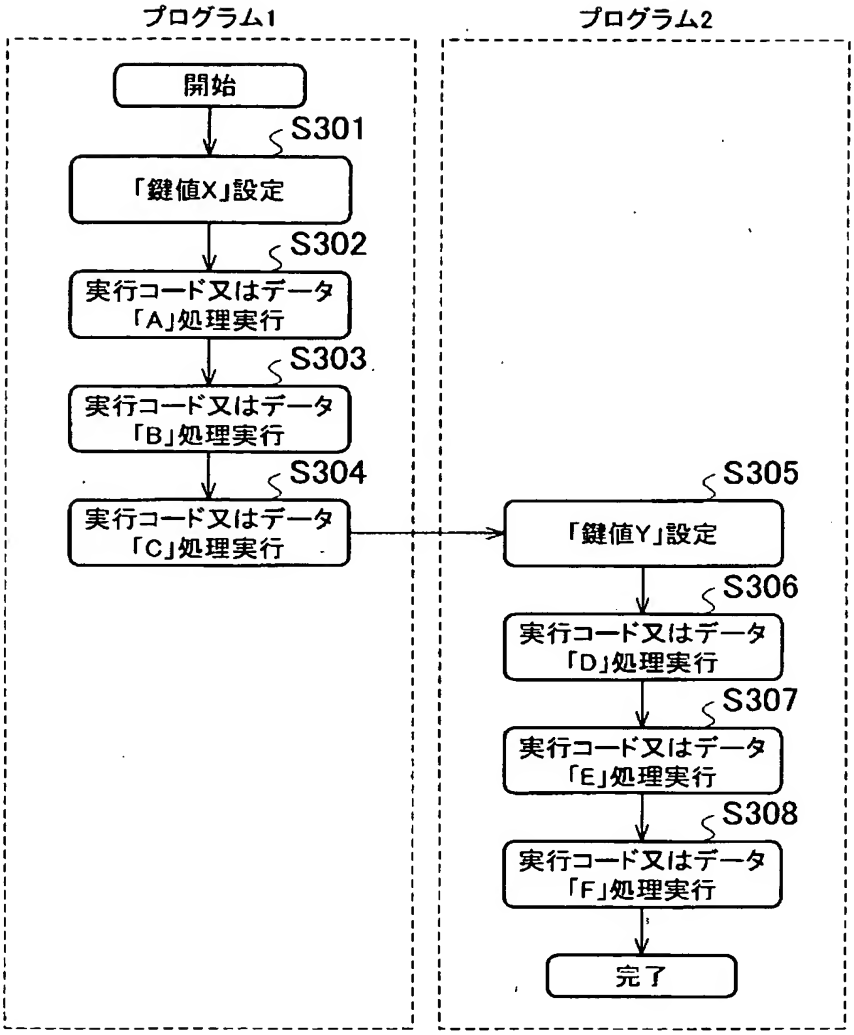
【図４】


開始
キャッシュメモリへの書き出し
S201
鍵値レジスタの暗号鍵を
秘密保護属性保持部に格納
S202
データをキャッシュメモリより取得
S203
暗号鍵を秘密保護属性保持部より取得
S204
データを暗号鍵により暗号化
S205
暗号化したデータを
外部メモリに書き出し
S206
完了

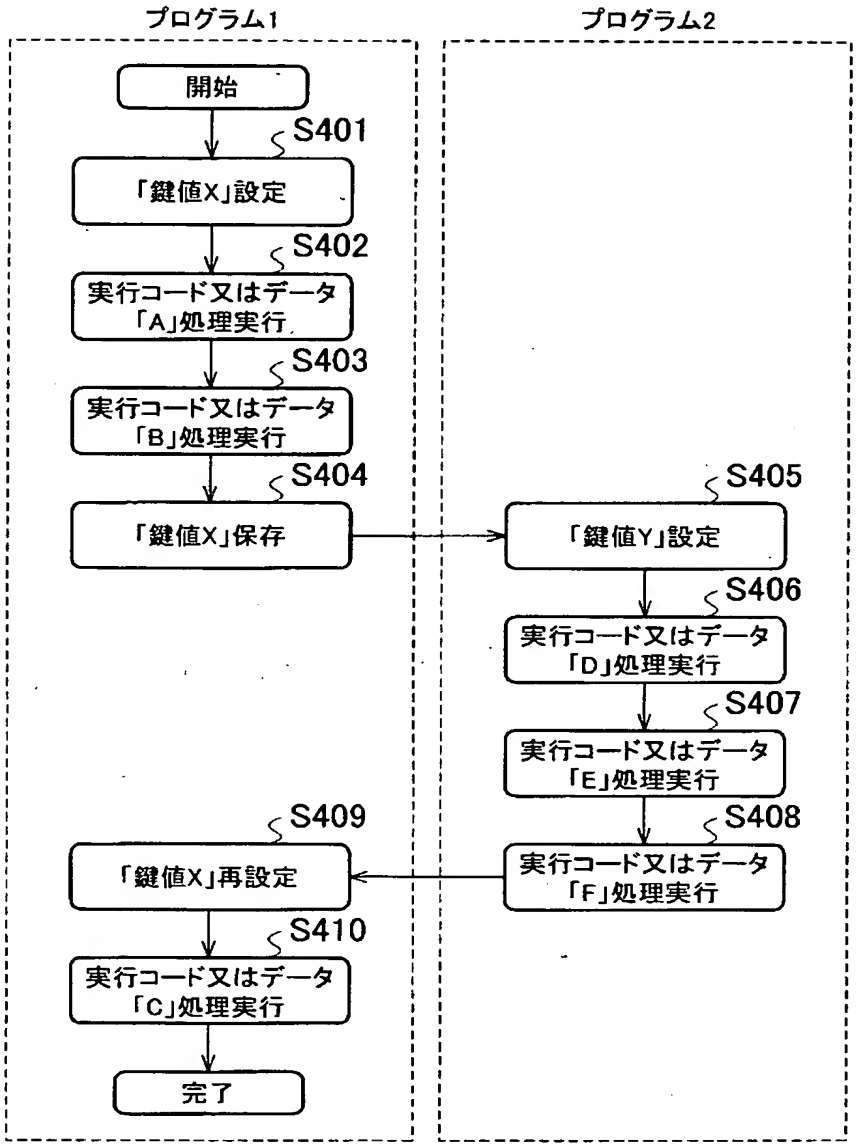
【図５】

プログラム	実行コード又はデータ	鍵
1	A	X
	B	X
	C	X
2	D	Y
	E	Y
	F	Y

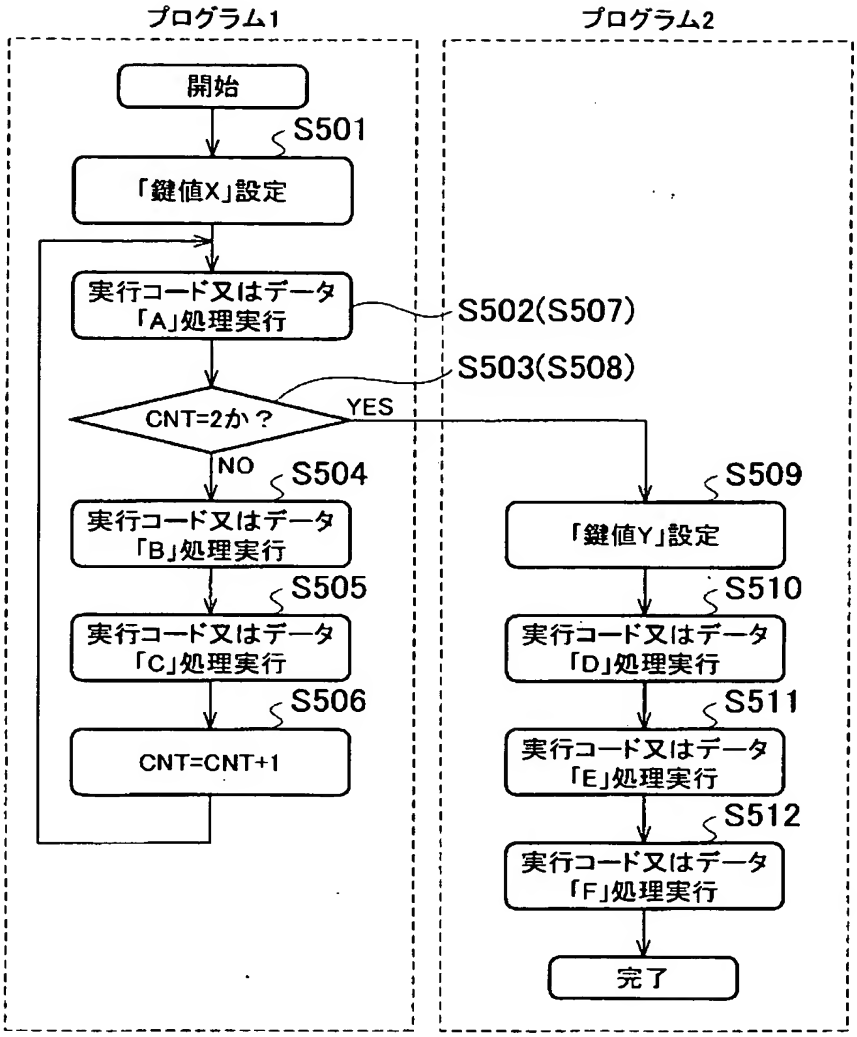
【図６】


プログラム1
プログラム2
開始
S301
「鍵値X」設定
S302
実行コード又はデータ
「A」処理実行
S303
実行コード又はデータ
「B」処理実行
S304
実行コード又はデータ
「C」処理実行
S305
「鍵値Y」設定
S306
実行コード又はデータ
「D」処理実行
S307
実行コード又はデータ
「E」処理実行
S308
実行コード又はデータ
「F」処理実行
完了

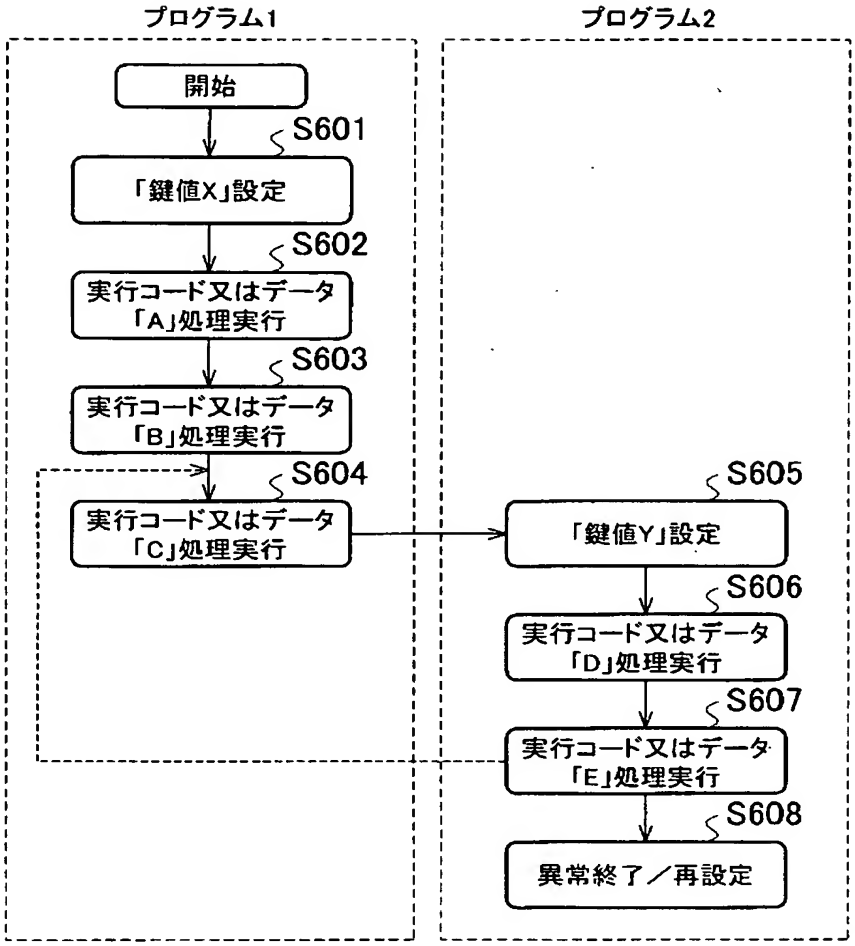
【図７】


プログラム1
プログラム2
開始
S401
「鍵値X」設定
S402
実行コード又はデータ
「A」処理実行
S403
実行コード又はデータ
「B」処理実行
S404
「鍵値X」保存
S405
「鍵値Y」設定
S406
実行コード又はデータ
「D」処理実行
S407
実行コード又はデータ
「E」処理実行
S408
実行コード又はデータ
「F」処理実行
S409
「鍵値X」再設定
S410
実行コード又はデータ
「C」処理実行
完了

【図８】


プログラム1
プログラム2
開始
S501
「鍵値X」設定
実行コード又はデータ
「A」処理実行
S502(S507)
S503(S508)
CNT=2か？
YES
NO
S504
実行コード又はデータ
「B」処理実行
S505
実行コード又はデータ
「C」処理実行
S506
CNT=CNT+1
S509
「鍵値Y」設定
S510
実行コード又はデータ
「D」処理実行
S511
実行コード又はデータ
「E」処理実行
S512
実行コード又はデータ
「F」処理実行
完了

【図９】


プログラム1
プログラム2
開始
S601
「鍵値X」設定
S602
実行コード又はデータ
「A」処理実行
S603
実行コード又はデータ
「B」処理実行
S604
実行コード又はデータ
「C」処理実行
S605
「鍵値Y」設定
S606
実行コード又はデータ
「D」処理実行
S607
実行コード又はデータ
「E」処理実行
S608
異常終了／再設定

【書類名】要約書

【要約】

【課題】　キャッシュメモリのアクセス制限がなく、キャッシュメモリの全領域を走査する必要がなく、複数のプログラムが１つの暗号鍵を共有する際安全な方法で他のプログラムから読み出すことができ、複雑なハードウェア機能が不要な耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法を提供する。

【解決手段】　耐タンパマイクロプロセッサ１００は、プロセッサコア１０、キャッシュメモリ制御部２０、コードデータ暗号復号処理部３０、鍵値レジスタ４０等を備える。鍵値レジスタ４０は暗号鍵を格納する。この暗号鍵は実行コード及びデータを記憶する為のキャッシュライン上の秘密保護属性保持部２５ａ～２５ｄにも格納される。キャッシュメモリ制御部２０は、鍵値レジスタ４０と秘密保護属性保持部２５ａ～２５ｄを比較することにより実行コード及びデータの同一判定処理を行う。

【選択図】　　図１

特願２００３－０１２５５８

出　願　人　履　歴　情　報

識別番号　　［０００００３０７８］

１．変更年月日　　２００１年　７月　２日
［変更理由］　　住所変更
住　所　　東京都港区芝浦一丁目１番１号
氏　名　　株式会社東芝

２．変更年月日　　２００３年　５月　９日
［変更理由］　　名称変更
住所変更
住　所　　東京都港区芝浦一丁目１番１号
氏　名　　株式会社東芝